

iOS integration documentation

1.Introduction:

The receipt printer adopts the ESC/POS commands commonly used in the printer industry. The iOS SDK encapsulates the low-power Bluetooth device discovery, connection, and data sending functions, which can be faster

to make. Commonly used ESC/POS commands have been encapsulated, such as font size, bold, down

Scribe, center; barcode printing; picture printing, etc.;

2. Glossary:

2.1 Print width

Refers to the maximum horizontal printing range that the printer can support, which is determined by the printer itself. example

For example, for a printer with a paper width of 80mm, the maximum effective width for printing is 72mm (576 dots),

For a 58mm paper width printer, the maximum effective printing width is 48mm (384 dots)

2.2. DPI

The number of dots printed in each inch, vertical or horizontal movement unit: default one

The unit of movement is a printing point, the unit of horizontal movement is 1/8mm, and the unit of vertical movement is 1/8mm. The unit is 1/8mm.

3. Integration instructions:

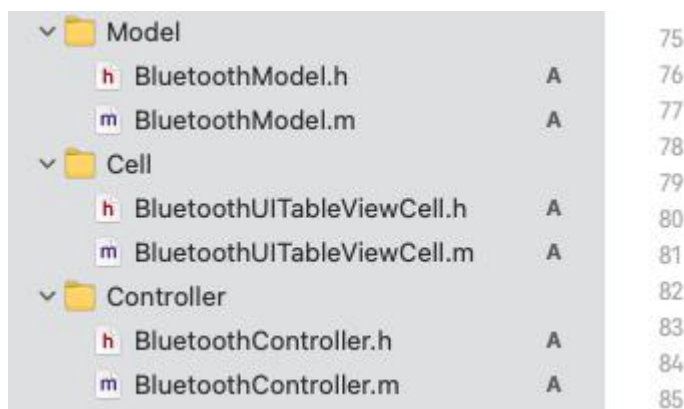
3.1 First Step:

Import the static library libPrintfESCLib.a and the header file into the project.



3.2 Second step:

Copy the files related to Bluetooth connection, including the data model BluetoothModel, the cell style BluetoothUITableViewController, and the view controller BluetoothController as shown in the following figure:



3.3 Third step:

After the completion of the second step above, it has been able to communicate with the printer via Bluetooth. The following is the style of controlling content sending, as shown below:

```
// 获取一个 ESC 控制器实例
    // Get an ESC controller instance
    PrintfESCManager* instance = [PrintfESCManager
createNew];

    // 清空数据
    // Clear data
    [instance clearData];

    // 初始化打印机
    // Initialize the printer
    [instance initPrinter];

    // 文本加粗
    // Bold text
    [instance setBoldMode:1];

    // 内容居中对齐
    // Align the content to the center
    [instance setSelectJustification:1];

    // 设置打印的文本
    // Set the printed text
    [instance setText:@"A good medicine tasks
bitter.Success often depends upon knowing how long it
will take to succeed.Wise men learn by other men's
mistakes;"];

    // 打印并走纸 6 行
    // Print and feed paper 6 lines
    [instance setPrintAndFeedRow:6];

    // 发送数据给打印机
    // Send data to the printer
    [self.printerBluetoothManager
writeNSData:[instance getData]];
```

4. PrintDemo Description:

The following is a brief description of how the Demo we provide discovers the device, connects to the device

To send data to the printer.

4.1. first step

First initialize the "PrinterBluetoothManager" instance, which encapsulates Bluetooth discovery, connection and other functions

```
//获取一个蓝牙管理实例

self.printerBluetoothManager =
[PrinterBluetoothManager
printerBluetoothManagerInstance];
```

4.2. second step

Set up the Bluetooth discovery callback and the callback of successful Bluetooth connection through the initialized instance of "PrinterBluetoothManager".

```
//发现蓝牙设备回调

[self.printerBluetoothManager
blePeripheralFound:^(CBPeripheral * _Nonnull
peripheral, NSNumber* rssi) {
    NSString *uuid =
peripheral.identifier.UUIDString;
    for(int i = 0; i <
weakSelf.bluetoothModels.count;i++){
        BluetoothModel *bluetoothModel =
weakSelf.bluetoothModels[i];
        NSString *tempUUID =
bluetoothModel.peripheral.identifier.UUIDString;
        if([uuid isEqualToString:tempUUID]){
            return;
        }
    }
}
```

```

        BluetoothModel *bluetoothModel =
[[BluetoothModel alloc] init];
        bluetoothModel.peripheral = peripheral;
        bluetoothModel.rssi = rssi;
        [weakSelf.bluetoothModels
addObject:bluetoothModel];
        [weakSelf.bluetoothModels
sortUsingComparator:^(NSComparisonResult(id _Nonnull
obj1, id _Nonnull obj2) {
            return [((BluetoothModel*)obj1).rssi
intValue] < [((BluetoothModel*)obj2).rssi intValue];
        }]);
        [weakSelf.bluetoothDeviceUITableView
reloadData];
    }];

    // 蓝牙连接成功回调

    [self.printerBluetoothManager
blePeripheralConnected:^(CBCentralManager * _Nonnull
central, CBPeripheral * _Nonnull peripheral) {
        NSLog(@"connect succeeded");
        [weakSelf dismissViewControllerAnimated:true
completion:nil];
    }];

```

4.3 third step

Use the initialized "PrinterBluetoothManager" to send data to the printer. The printed data is obtained through "PrintfESCManager".

1、 Print the text

```

    // 获取一个 ESC 控制器实例
    // Get an ESC controller instance
    PrintfESCManager* instance = [PrintfESCManager
createNew];

    // 清空数据
    // Clear data
    [instance clearData];

```

```

// 初始化打印机
// Initialize the printer
[instance initPrinter];

// 设置打印的文本
// Set the printed text
[instance setText:@"A good medicine tasks
bitter.Success often depends upon knowing how long it
will take to succeed.Wise men learn by other men's
mistakes.I wish I had a different life. I wish I was braver
and prettier or just happy. But it's useless to dream,
because nothing ever changes.The blues are because you're
getting far or maybe it's been raining too long."];
[instance setText:@"\n\n\n\n\n\n\n"];

// 发送数据给打印机
// Send data to the printer
[self.printerBluetoothManager
writeNSData:[instance getData]];

```

2、 Print barcode

```

PrintfESCManager* instance = [PrintfESCManager
createNew];
[instance clearData];
[instance initPrinter];
[instance setBarCodeWithType:BarCodeTypeCODE128
barcodeStringPosition:BarCodePositionTypeBELOW
barcodeHeightInDot:72 barcodeWidth:2
barcodeContent:@"No.123456"];
[instance setText:@"\n\n\n\n\n\n\n"];
[self.printerBluetoothManager
writeNSData:[instance getData]];

```

3、 Print image

```

UIImage *uiImage = [UIImage imageNamed:@"goodwork"];
int newWidth = 38 * 8;
uiImage = [self reSizeImage:uiImage
toSize:CGSizeMake(newWidth, (int)(newWidth *
uiImage.size.height / uiImage.size.width))];
PrintfESCManager* instance = [PrintfESCManager
createNew];
[instance clearData];
[instance initPrinter];

```

```
[instance setImage:uiImage left:0];  
[instance setText:@"\n\n\n\n\n\n\n"];  
[self.printerBluetoothManager  
writeNSData:[instance getData]];
```